WISM68: ENTORNO VISUAL EDITOR/ENSAMBLADOR/SIMULADOR PARA MC68000 - MANUAL DE USUARIO

Versión 0.1 - 18 de diciembre de 2001

ÍNDICE

1. INTRODUCCIÓN 1

2. INSTALACIÓN DE WISM68 1

3. ARRANQUE DE WISM68 1

4. EDICIÓN DE PROGRAMAS 2

5. ENSAMBLAJE DE PROGRAMAS 3

6. SIMULACIÓN DE PROGRAMAS 4

6.1. AYUDA 4
6.2. CARGA DE UN FICHERO OBJETO 4
6.3. VENTANAS DE VISUALIZACIÓN 4
6.3.1. Ventana de registros de la UCP 4
6.3.2. Ventana de memoria 4
6.3.3. Ventana de código desensamblado 5
6.4. PUNTOS DE RUPTURA 6
6.5. SIMULACIÓN DEL CÓDIGO 7
6.6. MANIPULACIÓN DE LA MEMORIA 8
6.7. INTERRUPCIONES 8

1. INTRODUCCIÓN

El programa descrito en este documento constituye un entorno visual editor-ensamblador-simulador que permite desarrollar código para el microprocesador Motorola 68000 desde un microcomputador de tipo PC compatible IBM con sistema operativo Windows. El programa puede obtenerse sin coste vía Internet en la página http://www.hildreds.freeserve.co.uk/WISM68/. El archivo, al menos en su versión actual, se llama WISM68_SETUP.ZIP.

2. INSTALACIÓN DE WISM68

Una vez descargado y copiado el archivo en nuestro disco duro, será preciso descomprimirlo con alguna utilidad que maneje archivos empaquetados ZIP (por ejemplo, XACE, que puede ser descargado gratuitamente desde http://www.winace.com).

Entre los archivos creados, hay uno que se llama SETUP.EXE. Al ejecutarlo aparecerá la ventana mostrada en la figura 1.

WISM68 Setup
Welcome to the WISM68 installation program.
Setup cannot install system files or update shared files if they are in use. Before proceeding, we recommend that you close any applications you may be running.
OK
Exit Setup

Figura 1. Ventana de inicio de instalación de WISM68.

Para rehusar la instalación será preciso recurrir a la opción Exit Setup. Si por el contrario pinchamos OK, aparecerá una ventana como la mostrada en la figura 2.

Si pinchamos sobre el botón grande del icono, instalaremos el simulador en la carpeta indicada. Si por el contrario preferimos instalarlo en otra carpeta distinta, pincharemos en el botón Change Directory, tras lo cual aparecerá una ventana que nos permitirá especificar una ubicación diferente para la aplicación.

3. ARRANQUE DE WISM68

Una vez instalado el programa, el menú Inicio contendrá una nueva entrada con el nombre WISM68. Seleccionar dicha entrada será suficiente para arrancar el programa. Tras ello aparece la ventana mostrada en la figura 3.

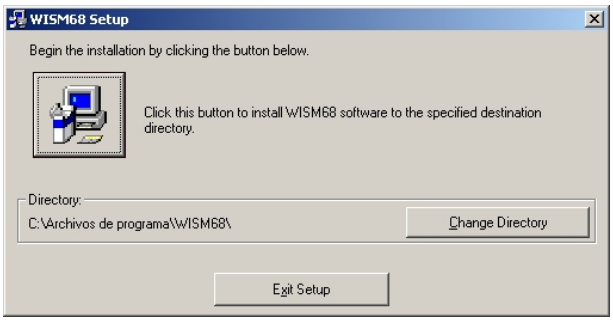

WISM68 Setup
Begin the installation by clicking the button below.
Click this button to install WISM68 software to the specified destination directory.
Directory:
C:\Archivos de programa\WISM68\
Change Directory
Exit Setup

Figura 2. Instalación de WISM68.

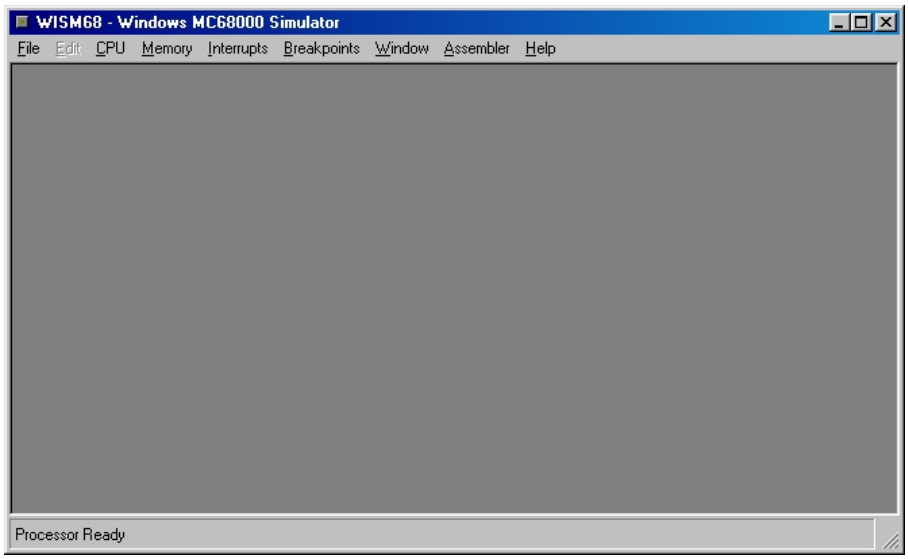

WISM68 - Windows MC68000 Simulator
File Edit CPU Memory Interrupts Breakpoints Window Assembler Help
Processor Ready

Figura 3. Pantalla principal de WISM68.

4. EDICIÓN DE PROGRAMAS

WISM68 contiene una ventana de edición. Para abrirla, será preciso ir a la opción File | Load Assembly Source, o pulsar simplemente CTRL-L. Aparece una ventana en la que podemos seleccionar el nombre del archivo. Al pulsar Aceptar se abrirá la ventana que mostrará el contenido de dicho archivo, y podremos modificar su contenido. Para guardar los cambios, será preciso utilizar la opción File | Save Assembly Source (CTRL-S) o File | Save Source As.

Las opciones del menú Edit también están disponibles para realizar la edición del código fuente.

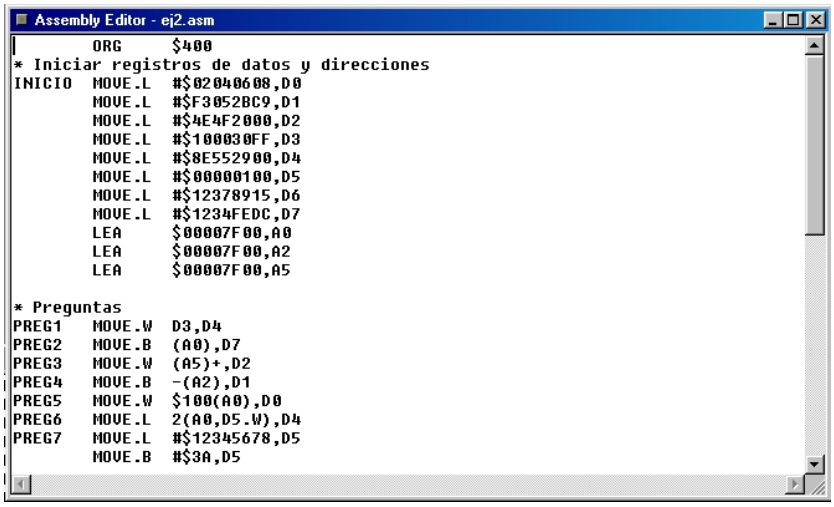


Figura 4. Ventana de edición.

5. ENSAMBLAJE DE PROGRAMAS

El código se ensambla eligiendo la opción Assembler | Assemble File (CTRL-A), tras lo cual se abre la ventana mostrada en la figura 5.

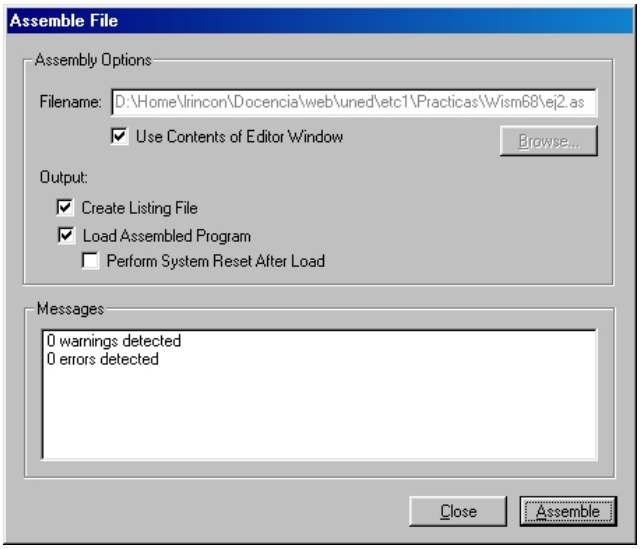


Figura 5. Ventana de ensamblaje.

En el cuadro Filename podemos decir el nombre del fichero fuente que vamos a ensamblar. Podremos seleccionar dicho fichero pinchando en Browse. Si activamos la opción Use Contents of Editor Window ensamblaremos el código presente en la ventana de edición.

La opción Create Listing File permite crear un fichero de listado. Por su parte, Load Assembled Program carga en el simulador el resultado del ensamblaje si no ha habido errores.

El ensamblaje se realiza pinchando sobre el botón Assemble. El cuadro Messages muestra información acerca del resultado del proceso, e indica si ha habido errores o no. En caso de no haberlos, se genera en disco un fichero llamado NOMBRE_FUENTE.H68 en formato de registros S de Motorola. Dicho fichero contiene el binario con el código objeto ya ensamblado y preparado para ser cargado en el simulador.

6. SIMULACIÓN DE PROGRAMAS

6.1. AYUDA

Desgraciadamente, en la versión actual de WISM68 la opción Help | Help Topics no funciona.

6.2. CARGA DE UN FICHERO OBJETO

Antes de proceder a la simulación de un programa, es preciso cargar el código objeto en el simulador. Para ello se puede proceder de dos formas:

- Ensamblar el código fuente activando la opción Load Assembled Program.
- Cargar un fichero previamente ensamblado. Para ello, será preciso ir a la opción File | Load Program y seleccionar el fichero pertinente.

6.3. VENTANAS DE VISUALIZACIÓN

El simulador dispone de varias ventanas que permiten ver los contenidos de los registros, la memoria simulada y los puntos de ruptura definidos (ver apartado 6.4). Para ello será suficiente con activar las entradas del menú Window. Podemos mantener todas las ventanas abiertas, o sólo algunas de ellas. Entre éstas se encuentra la ventana de edición, ya mostrada en el apartado 4.

6.3.1. Ventana de registros de la UCP

En dicha ventana (figura 6) podemos observar y modificar los contenidos de los registros del M68000, tanto los de datos como los de direcciones, punteros de pila y registro de estado.

6.3.2. Ventana de memoria

La ventana de memoria (figura 7) muestra los contenidos de la misma en hexadecimal (en la parte izquierda de la ventana) y en modo carácter (en la parte derecha). Nótese que lo primero que aparece en cada fila es la dirección de la primera posición de memoria mostrada (seguida del carácter ":").

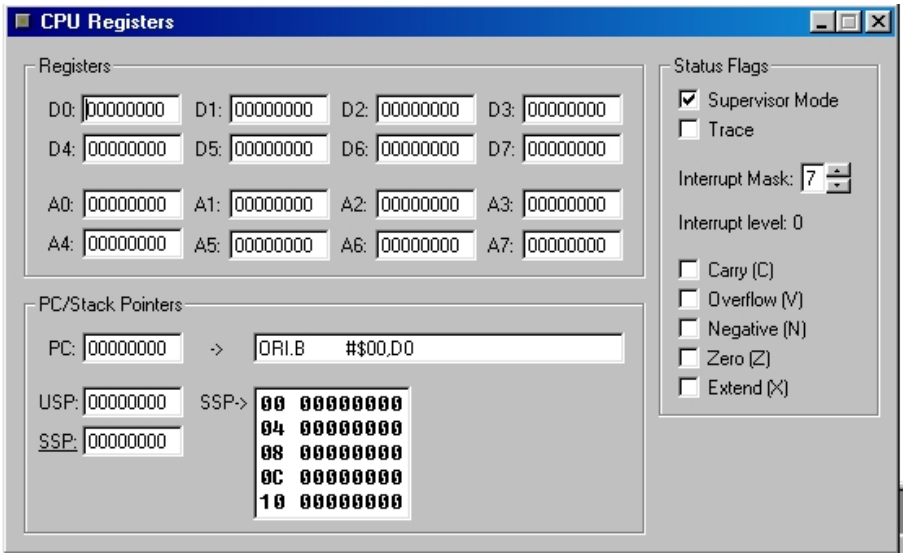


Figura 6. Ventana de registros de la UCP.

Es posible modificar el contenido de la misma seleccionando la ubicación que queramos e introduciendo el nuevo valor. Otra forma de alterar el contenido de la memoria es recurrir a las opciones del menú Memory (ver apartado 6.6).

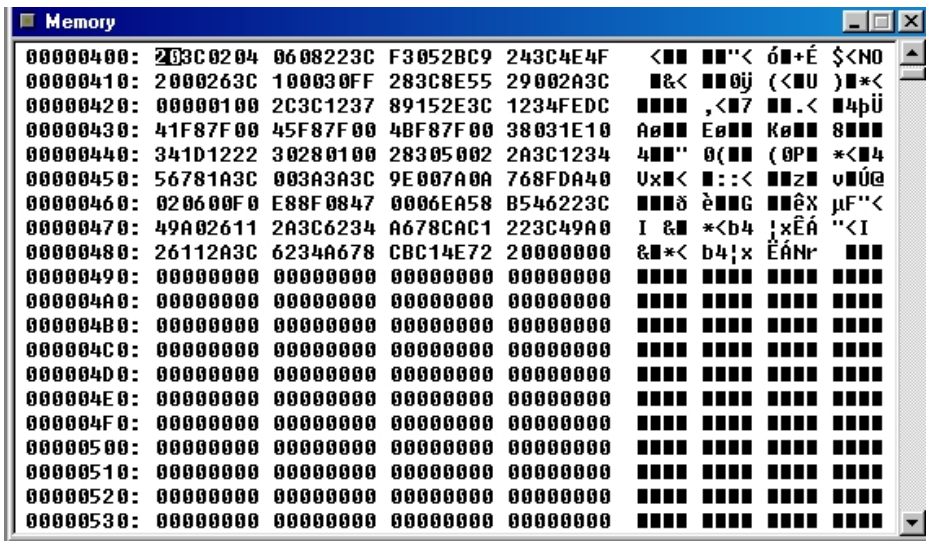


Figura 7. Ventana de memoria.

6.3.3. Ventana de código desensamblado

En esta ventana (figura 8) aparece la información almacenada en la memoria interpretando todo su contenido como instrucciones. Podemos seleccionar cualquier posición de memoria para comenzar el desensamblaje, o bien podemos indicar al simulador que desensamble a partir de la posición apuntada por el PC.

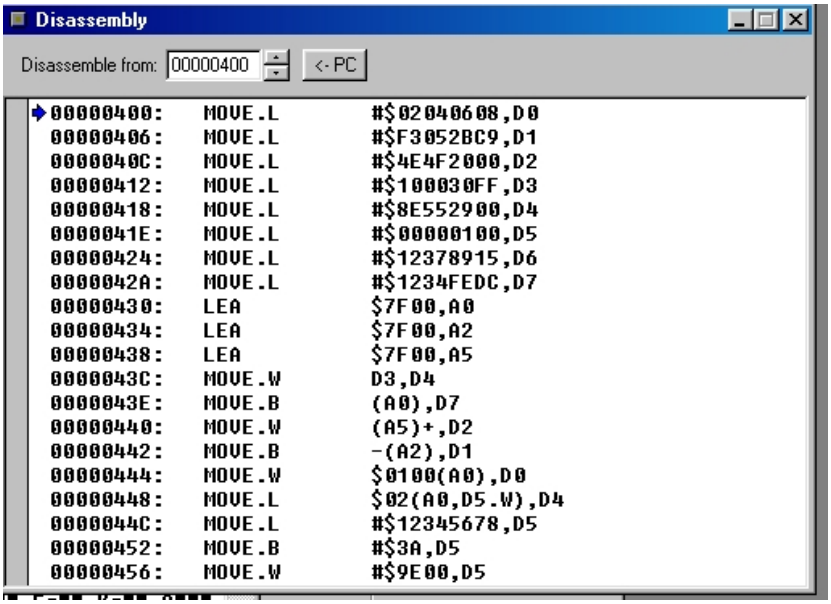


Figura 8. Ventana de código desensamblado.

6.4. PUNTOS DE RUPTURA

Los puntos de ruptura (*breakpoints*) permiten detener la simulación de instrucciones en puntos concretos del programa. Se utilizan para observar el comportamiento del programa en instrucciones determinadas sin tener que ir simulando todas las instrucciones del mismo una por una.

Para definir un punto de ruptura en el código será preciso recurrir a la opción Breakpoints | Set Breakpoint, tras lo cual se abre una ventana que permite indicar la posición de memoria donde se encuentra la instrucción en la que se quiere colocar el punto de ruptura (figura 9). Se puede indicar una condición adicional, con lo que el punto de ruptura se activaría únicamente si al llegar a dicha instrucción la condición antedicha se cumple. En ese caso nos encontraríamos ante un punto de ruptura condicional.

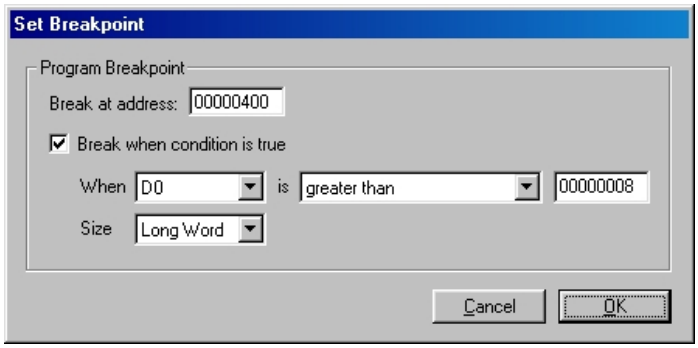


Figura 9. Definición de un punto de ruptura.

El menú contiene otra opción (Toggle Breakpoint) que permite poner o quitar un punto de ruptura en la posición apuntada por el contador de programa. La opción Clear All Breakpoints desactiva todos los puntos de ruptura.

Podremos ver dónde se encuentran definidos los puntos de ruptura en la ventana que aparece al seleccionar Window | Breakpoint (figura 10). Con la opción Edit podemos modificar sus propiedades. La opción Clear borrará el punto de ruptura.

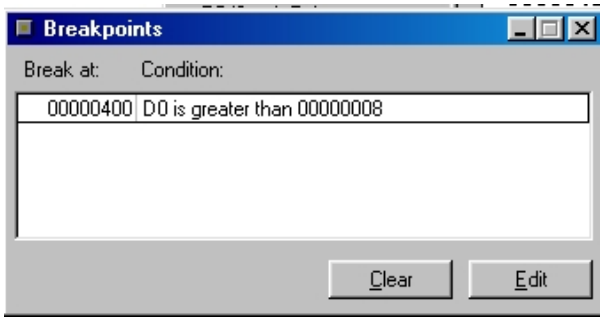


Figura 10. Ventana de puntos de ruptura.

También es posible definir puntos de ruptura actuando sobre la ventana de código desensamblado. Para ello, pincharemos con el ratón sobre la columna gris de la izquierda de dicha ventana, a la altura de la instrucción donde deseamos definirlo. En dicho lugar aparece una marca rectangular roja si el punto de ruptura está activado.

6.5. SIMULACIÓN DEL CÓDIGO

La simulación del código se efectúa a través de las opciones del menú CPU:

- Step (F1) ejecuta la instrucción apuntada por el PC. Si mantenemos abierta la ventana de registros o la de memoria, veremos que aparecen en rojo los elementos cuyo contenido cambia.
- Step Over (MAYÚS-F1): es parecido a Step, pero cuando llega una subrutina, la ejecuta completamente y se detiene tras retornar de la misma.
- Run (F2) ejecuta las instrucciones hasta llegar a una instrucción STOP o hasta el siguiente punto de ruptura.

Antes de proceder a la simulación del código, es preciso cerciorarse de que el PC apunte a la primera instrucción que se pretende simular. En la ventana de código desensamblado la instrucción apuntada por el PC aparece señalada con una flecha azul a la izquierda.

La opción Restore Default State pone los registros a 0 y borra el contenido de la memoria.

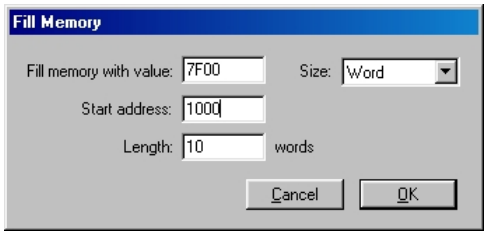
6.6. MANIPULACIÓN DE LA MEMORIA

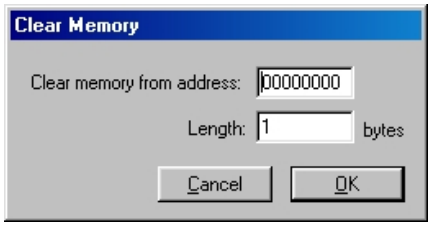
Podemos modificar el contenido de la memoria directamente sobre la ventana de memoria (ver apartado 6.3.2), o utilizando algunas de las opciones del menú Memory:

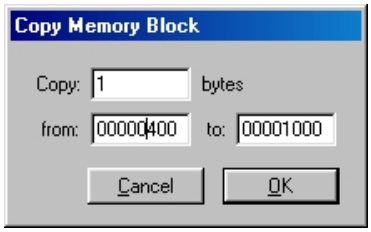
- Fill permite rellenar una zona de memoria con un dato determinado.
- Clear borra una serie de posiciones de memoria (las pone a 0).
- Copy Block puede usarse para copiar una zona de memoria en otra.

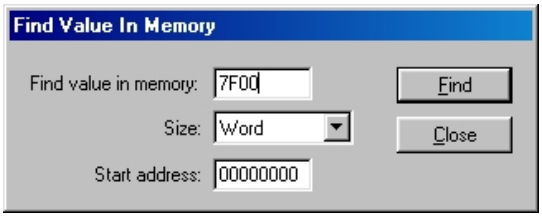
Otras opciones del mismo menú son útiles para realizar búsquedas en la memoria: Find permite indicar un dato y una dirección a partir de la cual buscarlo, y Find Next repite la búsqueda a partir de la última posición donde ha sido encontrado en la búsqueda anterior.

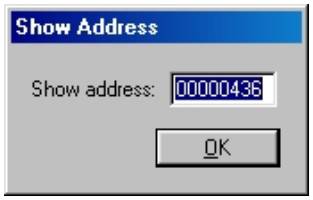
Show Address permite visualizar en la ventana de memoria los contenidos de la misma a partir de la posición que se le indique.











6.7. INTERRUPCIONES

El simulador es capaz de simular el comportamiento del MC68000 ante las interrupciones. La descripción de estas opciones queda postpuesta para una versión futura de este manual.